

# Android Beacon Library

## An Android library providing APIs to interact with Beacons

[Home](#) [Download](#) [Quick Start](#) [Sample Code](#) [Docs](#) [Support](#)

### Distance estimates

The API provides estimates of the distance to a beacon in meters. The values are rough estimates based on the signal strength of the Bluetooth signal received at the mobile device. Applications should not expect a high precision in the numbers. They are most useful for determining which beacon is closest when multiple are visible.

### How accurate are the estimates?

At close proximity of about 1 meter, you can expect to see distance estimates between 0.5-2 meters. At further distances you will see more variation. At 20 meters or actual distance, the estimate provided by the library may vary from 10-40 meters. The variation is caused by noise on the signal measurement, along with signal reflections and obstructions.

### Variations in time

In order to reduce noise on the estimate, the library averages signal measurements over 20 seconds, throws out the top 10% and bottom 10% of measurements, and averages the rest. It is important to understand that the distance estimates are based on the previous 20 seconds in time, so when a mobile device moves, the distance estimate will lag until the device has been stationary for 20 seconds, at which time the distance estimate will stabilize. The [variation in time is adjustable](#).

### Device variations

A big challenge in providing accurate distance estimates with Android devices is that every device model receives signals differently. Each model may have a different Bluetooth chipset and antenna, and therefore may receive a different signal level when in the same position relative to a beacon. In order to address this, the library uses a different formula for calculating distance for different Android device models. Because not all devices have custom formulas built into the library, the library will fall back to the default device calculation for the Nexus 5 if no matching formula is found.

### Formula

Despite formulas suggested by signal theory, the most accurate predictor of distance based on signal strength (RSSI) can be obtained by doing a power regression against a known table of distance/RSSI values for a specific device. This uses the formula  $d = A * (r/t)^B + C$ , where  $d$  is the distance in

meters,  $r$  is the RSSI measured by the device and  $t$  is the reference RSSI at 1 meter.  $A$ ,  $B$ , and  $C$  are constants.

## Data for specific models

The list of models with specific calculations is available in the config file [here](#). If you don't see your model listed, see the section below on how you can get it added

## Getting new Android device models added to the list

If you would like to get a new Android device model added to the list, you can help by taking measurements. In order to do this, you will need a beacon, an Android device, an iPhone 5, a copy of the Radius Networks' Android Locate app, and an accurate way of measuring distance between your mobile device and the beacon. You'll also need an outdoor space where you can do distance measurements that is free of obstructions in all directions for about 50 meters.

Once you have these tools, you can use the Calibration feature of the Android Locate app to get a 30 second running average of the RSSI for your device at specific distances. In order to do a new regression for your device, RSSI measurements must be collected at 20 different distances measured in meters: 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30.

You also need to collect the information about your Android device under test, which can be found under Settings -> About Phone. The fields needed are: Model number, Android version, Build number, Manufacturer.

Finally, you will need to measure the average RSSI seen by the iPhone 5 at a distance of 1 meter from the beacon. This is needed to provide an absolute reference for a known device.

You can see an example of the information collected for a Nexus 5 [here](#).

Once you collect this information, you can [calculate the formula constants](#).

© 2014, 2015 [Radius Networks](#)